BDTIC www.bdtic.com/Semiconductor

# Atmel PLDs' Architectures Simplify Timing Calculation(1)

## Introduction

This application note shows different graphical timing models that can help the user visualize the AC timing of the various Atmel PLD families of devices. Because of their deterministic and path-independent delays, timing calculation becomes straight forward.

Atmel PLDs have regular AND-OR architecture which simplifies timing calculation. All the AC timing parameters are clearly stated in the data book. Even for complex designs it only takes a few minutes to calculate the delays by hand.

If the design engineer has access to tools such as the Atmel-ViewPLD, he/she can easily predict the performance of the PLD. PLD software packages with timing simulation capabilities let the design engineer know the performance of the PLD immediately after the design is entered and check the results of the timing simulator and quickly modify the design to meet the system timing requirements. Atmel offers a complete design entry package called Atmel-ViewPLD that has such a timing simulator.

## Architectures/Timing Models

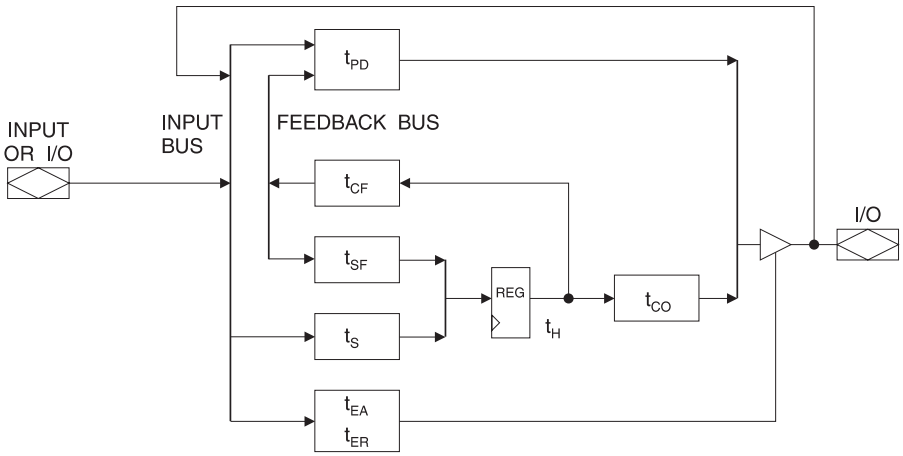
The AT22V10 represents the classic PAL™-type architecture with the programmable AND and fixed OR structure. A very small set of AC timing parameters can describe all the delays that occur in the implementation of register and combinatorial logic as shown in Figure 1.

For example, an output is described in the following Boolean equation:

OUTPUT = A0 & !A1 & A2 # SELECT;

Note: 1. Timing models for the ATF15XX devices are in their own specific application note (see: Using the ATF1500 CPLD).

**Figure 1.** AT22V10, ATV750, ATV750B, and ATV2500 Timing Model




Programmable Logic Device

Application Note

Rev. 0239D–08/99

A0, A1, A2, and SELECT can be inputs or I/O's and the delay will be simply the time for the signal to propagate from a pin through the AND-OR array, the macrocell, and to the output pin. This is described by the AC parameter $t_{PD}$.

The ATV750/ATV750B and ATV2500 with more advanced macrocells, maintain the same AND-OR structure as the AT22V10. Because of this, they can also be described by the same AT22V10 timing model. Even when using the buried registers found in the ATV750/ATV750B and ATV2500, the method of calculation for delays stays the same. For example, an ABEL™ description of a binary counter may look like:

```
COUNT.d = COUNT.fb + 1;
```

The counter can be implemented using only the internal buried registers of the ATV750, ATV750B, or ATV2500. In this case the minimum cycle time will be equal to $t_{CF}$ (clock to feedback) + $t_{SF}$ (feedback setup). See Figure 2. This is also equal to 1/($F_{MAX}$ internal).

Figure 3 shows the registered data path for a pin-to-pin delay, as might be described by output logic:

```
REG_A.d = A1 & B1 & !C1;
```

If A1, B1, and C1 are all signals from either input or I/O pins, then the minimum cycle time will be $t_S$ (setup time for input or I/O pin) + $t_{CO}$ (clock to output) which is equal to 1/($F_{MAX}$ external).

Figure 4 shows how data propagates for a typical Mealy state-machine, in which the state bits are inputs to combinatorial outputs:

```
COUNT.d = (COUNT.FB +1);
FULL = (COUNT.FB == ^HFF);
```

**Figure 2.** 1/($F_{MAX}$ internal)

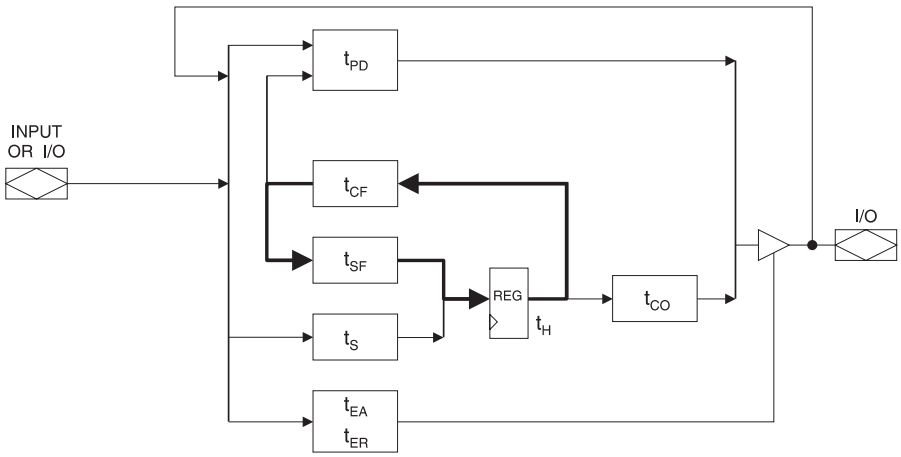
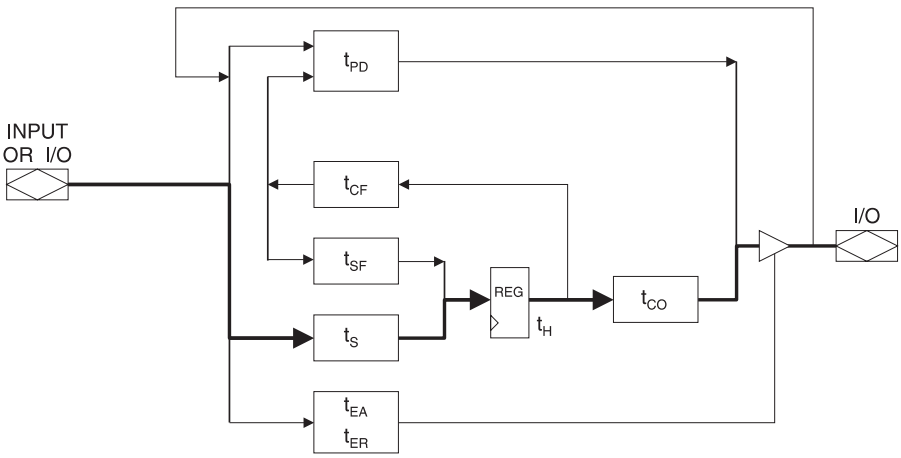



**Figure 3.** 1/($F_{MAX}$ external)



In this case, it will take FULL the delay of $t_{CF}$ + $t_{PD}$ to go from the rising edge of the clock driving the counter to the changing of FULL's output value.

These devices have both synchronous and asynchronous modes of operation. With the addition of the synchronous clocking option, the devices perform at a higher clock rate. The AC parameters have either the suffix of "S" (synchronous) or "A" (asynchronous) to distinguish the two registered clocking options.

Input latch setup and hold time are additional requirements when the latch is used. If the latch is bypassed, no delay is added.

For ATV2500B, the $t_{PD}$ parameter is broken down further to show different delay paths separately. $t_{PD1}$ and $t_{PD2}$ are similar to the traditional $t_{PD}$ parameter. $t_{PD1}$ is the delay from any pin to any combinatorial output. $t_{PD2}$ is the delay from internal feedback nodes to a combinatorial output pin.

$t_{PD3}$ is the delay from a pin to an internal combinatorial feedback.

$t_{PD4}$ is the delay from an internal feedback, through the AND/OR array, to an internal combinatorial feedback.

The ATV2500B lets the user configure internal combinatorial or registered feedbacks.

**Figure 4.** Mealy Machine Delay Path

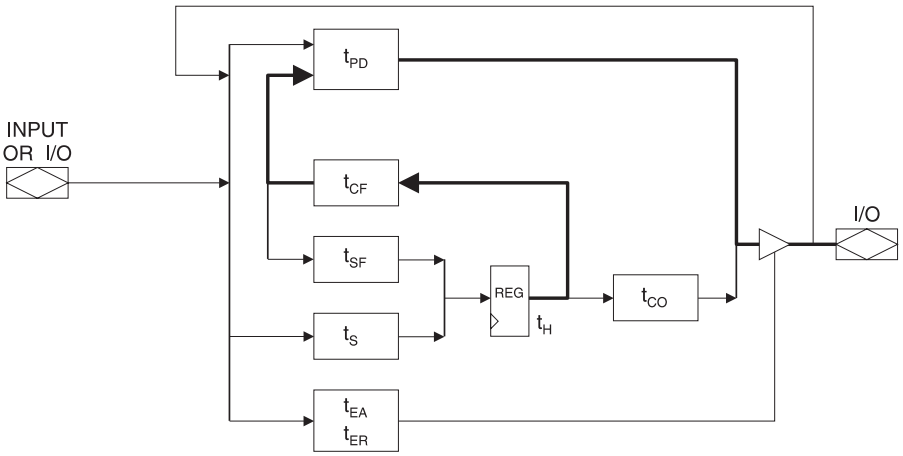
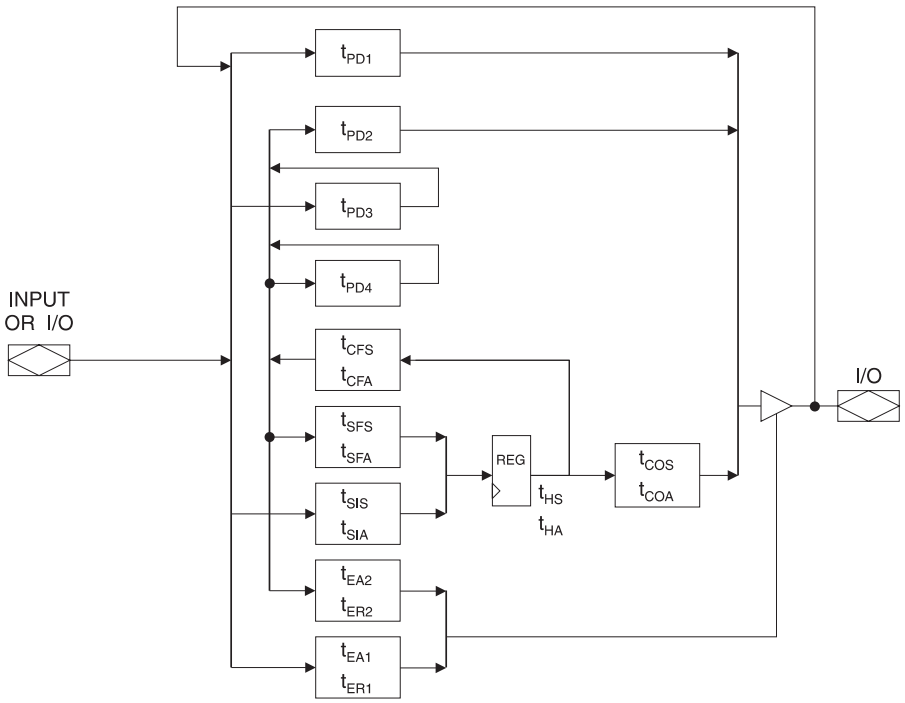



**Figure 5.** ATV2500B Timing Model



The most straightforward way to determine the delays is to look at the documentation generated by PLD software after the design has been reduced and fitted. If a reduced equation looks like:

OUTA = WATCHDOG “Product term 1

# A1 & A2 & A3 “Product term 2

# C_OUT; “Product term 3

1. Determine whether the logic is registered or combinatorial.
2. Determine whether OUTA is an internal node or an output pin.
3. Find the source of each of the components that makes up the product terms.

To analyze the various cases, let’s assume the following:

1. OUTA is combinatorial. We will look at OUTA implemented on an output pin versus OUTA implemented on a combinatorial node.
2. WATCHDOG is an internal registered node, A1 through A3 are directly from the inputs, and C_OUT is an internal combinatorial node (this covers all signal sources).

Table 1 summarizes the various timing requirements.

Case 1 (Figure 6) is the typical Mealy state machine where the internal state registers are decoded to form a combinatorial output. The total delay from clock to output is $t_{CF}$ + $t_{PD2}$.

Case 2 is the pin-to-pin delay. The AC parameter for that is $t_{PD1}$.

Case 3 is an internal combinatorial feedback’s delay from the AND/OR array to the output pin: $t_{PD2}$.

Case 4 is a “buried Mealy” where the internal state registers are decoded but not placed on an output pin. Instead the result is implemented on an internal combinatorial node where the logic is only useful internal to the design.

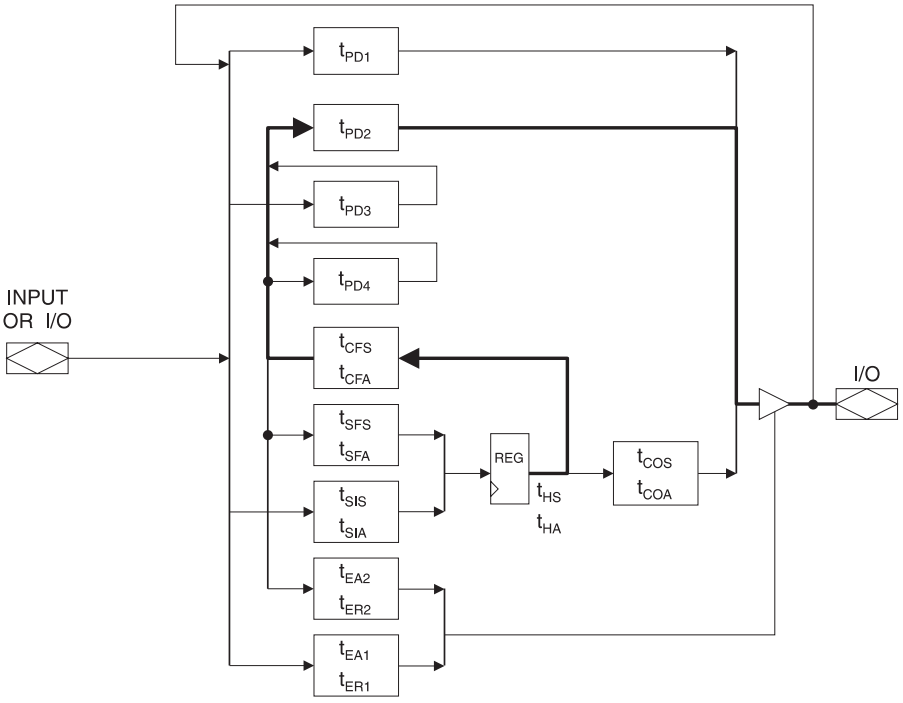
Case 5 is the delay from pin to internal combinatorial feedback: $t_{PD3}$.

Case 6 is the delay from one internal combinatorial node to another internal combinatorial node.

**Table 1.**

| | **Registered Feedback P.T. 1** | **Input from Pins P.T. 2** | **Internal Combinatorial Node P.T. 3** |
|---|---|---|---|
| Combinatorial Output OUTA | Case 1: $t_{CF}$ + $t_{PD2}$ | Case 2: $t_{PD1}$ | Case 3: $t_{PD2}$ |
| Combinatorial Node OUTA | Case 4: $t_{CF}$ + $t_{PD4}$ | Case 5: $t_{PD3}$ | Case 6: $t_{PD4}$ |

**Figure 6.** Typical Mealy State Machine in ATV2500B



## Conclusion

The graphical representation of the AC timing models illustrate how simple it is to determine the performance of logic implemented in a Atmel PLD. Atmel complex PLDs, even with their high pin counts and advanced features, have simple timing calculation. They aren't any harder to use than a common AT22V10.

BDTIC www.bdtic.com/Semiconductor